

US-PAT-NO: 6173345

DOCUMENT-IDENTIFIER: US 6173345 B1

TITLE: Method and apparatus for leveling transfer delays for a channel of devices such as memory devices in a memory subsystem

DATE-ISSUED: January 9, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Stevens; William A.	Folsom	CA	N/A	N/A
US-CL-CURRENT: 710/100; 711/167				

ABSTRACT: A method and apparatus for leveling transfer delays for a channel of devices. One method described determines a controller delay value by iteratively testing memory transfers to determine a largest transfer latency value using a subset of all available delays for at least one of a plurality of memory devices. Additionally, a memory device delay value for each of the plurality of memory devices is determined by testing memory transfers using at least one delay value for each of the plurality of memory devices.

51 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

----- KWIC -----

Abstract Text - ABTX: A method and apparatus for leveling transfer delays for a channel of devices. One method described determines a controller delay value by iteratively testing memory transfers to determine a largest transfer latency value using a subset of all available delays for at least one of a plurality of memory devices. Additionally, a memory device delay value for each of the plurality of memory devices is determined by testing memory transfers using at least one delay value for each of the plurality of memory devices.

Brief Summary Text - BSTX: Additionally, one or more delay registers may also be provided within individual RDRAM devices (e.g., a T.sub.RDLY register discussed in the Rambus "Direct RDRAM 64/72-Mbit" Data Sheet at p. 36). Values may be stored in these registers in order to equalize delays between the various devices along the channel. Typically, the controller delay value is initialized first, then the delay values for individual memory devices are adjusted.

Brief Summary Text - BSTX: The prior art may not provide a mechanism to reduce the number of cycles performed during initialization. In general, the prior art may not specify particular ways to test only a subset of the total number of possible delay values. The prior art also may not specify a method of choosing initial values for certain delay testing iterations, an efficient order for testing delay values, a method for performing each test, or a way of aborting delay testing and disabling devices when certain values are reached. Thus, the prior art may not provide an adequate method for leveling delays along a channel of devices.

Brief Summary Text - BSTX: A method and apparatus for leveling transfer delays for a channel

of devices is disclosed. One method disclosed determines a controller delay value by iteratively testing memory transfers to determine a largest transfer latency value using a subset of all available delays for at least one of a plurality of memory devices. Additionally, a memory device delay value for each of the plurality of memory devices is determined by testing memory transfers using at least one delay value for each of the plurality of memory devices.

Detailed Description Text - DETX: The disclosed techniques may provide an efficient initialization process for leveling delays between a controller and a set of devices. The initialization process may be efficient because only a subset of the total number of possible delay values need to be tested. For example, exhaustive delay testing may be avoided by using an efficient order for testing delay values, by choosing particular initial values for certain delay testing iterations, and/or by aborting delay testing when certain values are reached. As a result of utilizing the disclosed techniques, other processing may commence or restart more rapidly, thereby allowing increased system throughput.

Detailed Description Text - DETX: Efficient initialization of a memory subsystem may be particularly difficult due to a lack of storage for variables during initialization. Since the memory itself is being initialized, it may not be available for storage of temporary values. Additionally, the basic input/output system (BIOS) program which typically performs the initiation usually has little or no memory in which to temporarily store values. Thus, an initialization routine may have limited resources available to support an optimal initialization flow.

Detailed Description Text - DETX: Some embodiments of the disclosed two phase technique may avoid the need for large amounts of storage by first testing for a maximum latency to set a controller delay value and then cycling through each memory device to set a device delay value. Efficiency may be maintained by intelligently cutting out unnecessary iterations in both phases of the process.

Detailed Description Text - DETX: FIG. 1 illustrates a generalized flow diagram for delay leveling for a set or a channel of devices which exchange data with a controller over a bus. Phase 1 of this technique involves programming a controller delay value based on a maximum device latency. The controller delay value determined in phase 1 is a function of the largest transfer latency value of the devices on the bus. Depending on the particular transfer initiation and/or data receipt hardware employed, the controller delay value may either be equal to or derived from the largest transfer latency value.

Detailed Description Text - DETX: One possible technique of finding the controller delay determined by iterative testing. As indicated in block 100, however, a controller delay value may be determined by testing only a subset of delay values available. In other words, a controller may have N possible internal delay values which can be used to compensate for delays of the various devices along a bus, and less than N tests will be run on at least some of the memory devices under certain conditions. This testing of a subset of delay values may be accomplished, for example, by aborting a phase of the testing when a certain maximum controller delay value is found or by starting new iterations at the final value determined for a previous device.

Detailed Description Text - DETX: Operation of one embodiment of the system of FIG. 2 is illustrated in the flow diagrams of FIGS. 3 and 4 which respectively illustrate phase 1 and phase 2 operations. In block 300 of FIG. 3, phase 1 begins. In block 305, some variables are initialized. The memory controller delay value is initialized to an initial value that represents the number of delay cycles first tested by the memory controller in determining the final controller delay value. In one embodiment, zero is chosen as the initial value to ensure that the controller delay value is no greater than necessary.

Detailed Description Text - DETX: If the transfer is not successful, the memory controller delay is incremented as indicated in block 325 to give the data more time to reach the memory controller. Whether the memory controller delay exceeds a maximum delay value supported by the controller (e.g., in the embodiment of FIG. 2, four), as tested in block 350. If so, the controller is unable to insert enough delays to read the data from the device, and no further tests are performed on that device as the serial ID is decremented in block 340. In one embodiment, the maximum delay value is four.

Detailed Description Text - DETX: If, on the other hand, the delay is less than the maximum delay value supported by the memory controller, the memory controller 200 may delay the clock signal from the variable delay 210 by another cycle and again try to capture the data. Thus, the process returns to block 315 if the controller delay is not greater than the maximum delay supported by the controller as tested in block 350.

Detailed Description Text - DETX: Once a successful transfer is detected in block 320, the proper controller delay has been determined for that particular memory device as indicated in block 330. If the delay is the maximum delay supported by the controller, as tested in block 335, phase one is completed as indicated in block 355. Phase 1 is completed because the controller must utilize its maximum delay value for this memory device and there is accordingly no further need to test other memory devices. Thus, exhaustive controller delay testing may be aborted to save initialization time.

Detailed Description Text - DETX: If the controller delay value is not the maximum delay value, then other memory devices are checked to determine their delay values. As indicated in block 340, the serial ID counter is decremented to select a next memory device. In block 345, it is determined whether there are any more memory devices to be tested. If not, phase one ends as indicated in block 355. In the embodiment where the highest ID number is first selected and subsequent devices are selected by decrementing the ID number, the determination of whether there are any more memory devices to be tested may be made by testing whether the ID number is less than zero. Alternative tests may be used for different ID numbering and/or test sequencing.

Detailed Description Text - DETX: If there are any more memory devices to be tested, the process returns to block 310. The controller delay value determined to be sufficient for the prior memory device is the starting point for the next iteration. Even if the new memory device returns data more quickly, the memory controller is nonetheless limited by the slower previous device and may not be able to decrease the controller delay value. Therefore, by starting with the previously

determined controller delay value, the number of testing cycles may be reduced and the initialization process shortened.

Detailed Description Text - DETX: Phase 2 begins in block 400 of FIG. 4. Again variables are initialized as indicated in step 405. The serial ID counter is initialized to the serial ID of the last device on the channel, making the last device the selected device to be tested. The device delay for the selected device is set to an initial value. In one embodiment, zero may be used as an initial value since some devices may operate properly with no additional delay.

Detailed Description Text - DETX: As indicated in block 410, a memory transfer is conducted to test the selected memory device. If the transfer attempted in block 410 is not successful, as tested in block 412, the tested device delay value is compared to a maximum device delay value as indicated in block 415. If the tested device delay value is the maximum delay value, then the device has failed to accurately transmit data to the memory controller despite adding the maximum delay to its data output. In this case, the memory device is disabled, as indicated in block 425. The device may be disabled by assigning a device ID which is not used by the memory controller so that data transfers between the disabled device and the controller do not occur. Thereafter, other devices may be tested as the serial ID counter is advanced and the process continues with block 425.

Claims Text - CLTX: determining a memory device delay value for each of the plurality of memory devices by testing memory transfers using at least one delay value for each of the plurality of memory devices.

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by a controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ; and

Claims Text - CLTX: initializing a controller delay value to an initial delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device;

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value; and

Claims Text - CLTX: if the controller delay value is less than a maximum controller delay value, then

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ; and

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value, then selecting a new memory device as the selected memory device unless the controller delay value has been tested for all of the plurality of memory devices.

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by a controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ; and

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value in a register of the controller; and

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device.

Claims Text - CLTX: determining an individual memory delay value for each of a plurality of memory devices, the individual memory delay value being a number of delays up to a maximum delay value which levelizes a read delay for the plurality of memory devices; and

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ;

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device by assigning an unused device identification number;

Claims Text - CLTX: if the device delay value is less than the maximum device delay value then testing whether another device delay value for another memory device is less than the maximum

device delay value .

Claims Text - CLTX: determining a memory device delay value for each of the plurality of memory devices by testing memory transfers using at least one delay value for each of the plurality of memory devices.

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by the memory controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ; and

Claims Text - CLTX: initializing a controller delay value to an initial delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device;

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value; and

Claims Text - CLTX: if the controller delay value is less than a maximum controller delay value, then

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ; and

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value, then selecting a new memory device as the selected memory device unless the controller delay value has been tested for all of the plurality of memory devices.

Claims Text - CLTX: determining a memory device delay value for each of the plurality of memory devices by testing memory transfers using at least one delay value for each of the plurality of memory devices.

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by a controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ; and

Claims Text - CLTX: initializing a controller delay value to an initial delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device;

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value; and

Claims Text - CLTX: if the controller delay value is less than a maximum controller delay value, then

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ; and

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device;

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value, then selecting a new memory device as the selected memory device unless the controller delay value has been tested for all of the plurality of memory devices.

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by a controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ; and

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value in a register of the controller; and

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device.

Claims Text - CLTX: determining an individual memory delay value for each of a plurality of memory devices, the individual memory delay value being a number of delays up to a maximum delay value which levelizes a read delay for the plurality of memory devices; and

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ;

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device by assigning an unused device identification number;

Claims Text - CLTX: if the device delay value is less than the maximum device delay value then testing whether another device delay value for another memory device is less than the maximum device delay value .

Claims Text - CLTX: determining a first delay value for a first memory device by iteratively increasing the first delay value from an initial delay value, the first delay value being a number of cycles delay until data is received by a controller from the selected memory device;

Claims Text - CLTX: determining a next delay value for a next memory device starting at the first delay value ;

Claims Text - CLTX: testing whether the controller delay value is a maximum controller delay value ; and

Claims Text - CLTX: if the controller delay value is the maximum controller delay value then:

Claims Text - CLTX: storing the maximum controller delay value as the controller delay value in a register of the memory controller; and

Claims Text - CLTX: if the controller delay value is less than the maximum controller delay value then returning to testing whether the transaction to the selected memory device succeeds after selecting a new memory device as the selected memory device.

Claims Text - CLTX: determining an individual memory delay value for each of a plurality of memory devices, the individual memory delay value being a number of delays up to a maximum

delay value which levelizes a read delay for the plurality of memory devices; and

Claims Text - CLTX: testing whether a device delay value for a memory device is less than a maximum device delay value ;

Claims Text - CLTX: if the device delay value is greater than the maximum device delay value then disabling the memory device by assigning an unused device identification number;

Claims Text - CLTX: if the device delay value is less than the maximum device delay value then testing whether another device delay value for another memory device is less than the maximum device delay value .